AN ITERATIVE TECHNIQUE FOR AN ANALYTICAL SOLUTION TO TWO-POINT BOUNDARY PROBLEMS

by

VASANT MADHAVRAO PANDAV

B. E., Marathwada University, Aurangabad, India, 1964

A MASTER'S THESIS

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Mechanical Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1967

Approved by:

Hugh L. Walker
Major Professor

TABLE OF CONTENTS

LIST OF TABLES

CHAPTER I

INTRODUCTION

When a practical problem in science or technology permits mathematical formulation, the chances are rather good that it leads to one or more differential equations. This is certainly true of the vast catagory of problems associated with beams and columns, heat transfer and fluid flow, elasticity and electricity, etc.

Many of these problems which can be neatly formulated as differential equations can go no further for lack of solutions. At this point one discovers how few, relatively speaking, are the equations that have solutions in closed form.

Faced with this situation, various numerical methods have been devised that squeeze the desired information out of the differential equation directly. However, most of these methods yield a solution over a limited range of the domain, and this is in the form of a table, giving values of the dependent variable and sometimes its few derivatives for specific values of the independent variable.

Many times this defines the solution well enough, particularly if the solution has been tabulated with a sufficiently small increment of the independent variable. But however fine the increment be, the solution does not compare with an analytical

one, in that it cannot be handled analytically - put in equations, differentiated or integrated as desired, and so on.

Iterative methods, improving an assumed analytical solution with each step, have been suggested, and successfully applied to problems of initial-value type and also to those boundary value problems for which Green's function is known. Iteration was first applied to a technical eigenvalue problem in 1898 by L. Vianello [21]* in a study of buckling problems. The process was applied by A. Stodola [18] in 1904 to the problem of critical speeds of rotating shafts. The theory of the method had already been presented by H. A. Schwarz [16] in 1885 and been developed by E. Picard [13]. It is interesting to note that all these early works dealt with a particular class of eigenvalue problems. No effort was ever made to apply the technique to equilibrium problems and not many instances could be cited when an attempt was made to develop this technique further to apply to a wider class of problems.

It has been attempted in the present work to develop a technique for obtaining an analytical solution to boundary value problems governed by quasi-linear differential equations. The solution was intended to be a power series in the independent variable. The treatment here was limited to problems in one dimension, which include the initial value problems as a subclass. However, the author has a feeling that with enough modifications, the technique could be extended to problems in more than one dimension.

*Numbers in brackets refer to references in Bibliography

CHAPTER II

THE PROBLEM

Nomenclature

English alphabets:

A area of cross-section

b breadth of section

B coefficient matrix in the matrix equation representing linear boundary conditions

B_i ordinary differential operator

c_i coefficients in the series expansion of the solution function

C column vector of c_i, i = 1, 2,..., m.

d depth of section, d_1, d_2 -- end depths

D domain of definition; d/dx; coefficient matrix in DC = F (refer p. 13)

E modulus of elasticity

F constant column vector in DC = F (refer p. 13)

G ordinary differential operator

I moment of area of cross-section

K constant column vector in the matrix representation of linear boundary conditions

L length

m order of the governing differential equation

M quasi-linear ordinary differential operator, containing the highest order derivative term

n number of boundary conditions at the left end

N ordinary differential operator

x the independent variable

y the dependent variable

Greek alphabets:

α ratio of end depths = d_2/d_1; $\lambda^{1/4}$

β $\lambda^{1/4}/L$

γ $(\cos\alpha - \cosh\alpha)/(\sinh\alpha - \sin\alpha)$

λ eigenvalue, in the non-dimensional formulation

ρ density

$\sum$ summation sign

ω angular frequency

Subscripts and superscripts:

Numberical subscript to y or λ represent that particular mode.

Roman superscript to y represents differentiation.

Numberical or alphabetic superscript in parentheses represent a particular iteration.

Statement of the Problem

A boundary value problem, in general, may be stated as

$$\begin{aligned} M[y] &= \lambda N[y] && \text{in } D \\ B_i[y] &= 0 && i = 1, 2, \ldots, m \end{aligned} \qquad \text{(II-1)}$$

on the boundary of D.

Here D is the domain of definition of the problem and, in case of two-point boundary value problems, is a one-dimensional continum. M and N are ordinary differential operators, M being quasi-linear in nature. The order of the differential equation m is the same as that of the operator M and is larger than the order of the operator N.

The presence or absence of λ, an undetermined parameter, determines whether the problem belongs to the eigenvalue class or equilibrium class. In the case of eigenvalue problems the boundary conditions are essentially homogeneous. The formulation is satisfied by an infinity of values of λ and the corresponding eigenfunctions y.

In the present work a solution of the formulation was sought in the shape of y expressed as power series in the independent variable x. In case of eigenvalue problems, the first few eigenvalues and the corresponding mode shapes constituted what was expected as solution.

The Technique

The governing differential equation of the formulation, being quasi-linear in nature, was very easily solved for the highest derivative of y.

$$D^m y = G[y]$$

$$\text{or } y = D^{-m}(G[y]) + \sum_{i=1}^{m} c_i \, x^{i-1} \qquad \text{(II-2)}$$

where D = d/dx and the negative powers indicate integration. The c_i, the constants of integration, were determined by the use of the m boundary conditions -

$$B_i[y] = 0 \qquad i = 1, 2, \ldots, m.$$

An iterative process was then set up

$$y^{(r+1)} = D^{-m}(G[y^{(r)}]) + \sum_{i=1}^{m} c_i^{(r+1)} x^{i-1}$$

$$B_i[y^{(r+1)}] = 0 \qquad \text{(II-3)}$$

$$r = 1, 2, \ldots.$$

To start with, a polynomial was selected as an initial guess for y i.e. $y^{(0)}$. Any polynomial would serve the purpose but it is advisable to select one that satisfies all the boundary conditions. The proper choice of $y^{(0)}$ would certainly accelerate the convergence of the process.

This was enough to proceed, if the parameter λ did not enter the formulation. Otherwise, it sometimes necessiated a guess for

λ (in addition to that of solution function) at every step of iteration. However, this was easily furnished by Raleigh's quotient of the trial function.

It was observed that the iteration process here built up a power series approxitation to the true solution, each iteration adding substantially a few terms to the expansion (usually not less than the order of the differential equation, m.). If the process was to converge (as it did and could be expected to in most of the well behaved cases) after enough iterations the first few terms would quit changing. The convergence could be observed by comparing the successive iterates.

In eigenvalue problems the process converged to the mode corresponding to the smallest eigenvalue. The orthogonality condition was used to extract the higher modes.

Convergence of the Iteration

For initial-value problems, the scheme is the same as the extension of Picard's method for higher order equations. The proof has been exhibited in the references [15], [12]. Levy and Baggott [8] give a proof of convergence for systems with second order differential equations of both initial-value and equilibrium type. A general proof to cover eigenvalue problems cannot be given. However, a proof for convergence of an iteration procedure in general is given by Collatz ([2], pp. 36-48). Also, the convergence of an iteration procedure, employing the inverse of the operator M, is suggested in reference ([3], p. 303). When M is simply D^m, the method presented in the thesis coincides with the method mentioned above. The proof is given below:

The eigenvalue problem

$$\begin{aligned} M[y] &= \lambda N[y] \\ B_i[y] &= 0, \qquad i = 1, 2, \ldots, m \end{aligned} \tag{II-4}$$

can be written as

$$y = \lambda G[y]$$

where $G = M^{-1}N$.

Assume that the expansion theorem holds, and also that $\lambda_1 < \lambda_2 \leq \lambda_3 \leq \lambda_4 \leq \cdots$ Any admissible function $y^{(0)}$ can then be expressed as a linear combination of the individual modes y_i.

$$y^{(0)} = c_1 y_1 + \sum_{i=2}^{\infty} c_i y_i$$

Starting from an arbitrary admissible function $y^{(0)}$, an iteration process is set up according to the recurrence relation

$$y^{(n+1)} = G[y^{(n)}]$$

$$B_i[y^{(n+1)}] = 0$$

$$n = 1, 2, 3, \text{------}$$

For an eigenfunction y_i

$$G[y_i] = y_i/\lambda_i \ .$$

$$y^{(1)} = G[y^{(0)}]$$

$$= G[c_i y_i + \sum_{i=2}^{\infty} c_i y_i]$$

$$= \frac{c_1}{\lambda_1} y_1 + \sum_{i=2}^{\infty} \frac{c_i}{\lambda_i} y_i$$

$$= \frac{1}{\lambda_1}(c_1 y_1 + \sum_{i=2}^{\infty} \frac{\lambda_1}{\lambda_i} c_i y_i)$$

$$y^{(2)} = G[y^{(1)}]$$

$$= G[\frac{1}{\lambda_1}(c_1 y_1 + \sum_{i=2}^{\infty} \frac{\lambda_1}{\lambda_i} c_i y_i)]$$

$$= \frac{1}{\lambda_1}(\frac{c_1}{\lambda_1} y_1 + \sum_{i=2}^{\infty} \frac{\lambda_1}{\lambda_i} \frac{c_i}{\lambda_i} y_i)$$

$$= \frac{1}{\lambda_1^2}(c_1 y_1 + \sum_{i=2}^{\infty} (\frac{M}{\lambda_i})^2 c_i y_i)$$

Continuing in the same manner,

$$y^{(n)} = \frac{1}{\lambda_1^n}\left(c_1 y_1 + \sum_{i=2}^{\infty} \left(\frac{\lambda_1}{\lambda_i}\right)^n c_i y_i\right).$$

As n gets larger, the quotient $\left(\frac{\lambda_1}{\lambda_i}\right)^n$ tends to zero. So that,

$$y^{(n)} = \frac{1}{\lambda_1^n} c_1 y_1 \qquad \text{(II-5)}$$

and $$\lambda_1 = \frac{y^{(n)}}{y^{(n+1)}}.$$

Thus the iteration converges to the eigenfunction corresponding to the smallest eigenvalue. In general, the process will converge to the eigenfunction corresponding to the first non-zero c_i, if carried out exactly. While working approximately, however, small components of y_1 will be inevitably introduced, and the process will finally converge to the eigenfunction corresponding to the smallest eigenvalue independent of the choice of $y^{(0)}$.

CHAPTER III

COMPUTERIZATION

It is difficult to estimate in advance how much computation will be required to obtain a solution by iteration. The amount of computation per cycle increases with the number of cycles. The number of cycles required depends on the accuracy desired and on the particular system to be solved. So that, at some stage, hand computation goes out of the question and recourse has to be taken to some high-speed computing machine. The problems illustrated in the present work were programmed for IBM 1410/7010 Operating System (1410-PR-155) FORTRAN-1410-FO-970.

Standardization

For adapting the technique to digital computers, standardization of the formulation was done. First the formulation was reduced to non-dimensional form. One of the two boundary points was made to coincide with the zero of the independent variable axis (x-axis), and the scale of the variable x was adjusted so that the separation between the boundaries was unity, unless otherwise necessary.

Boundary Conditions

Each iteration involved a m-fold integration and hence gave rise to m constants of integration. These constants were to be determined so as to make the iterate satisfy the boundary conditions. Application of each condition resulted in an algebraic relation in the m constants of integration to be determined. Thus a set of m algebraic equations was obtained. The complexity of these equations naturally depended upon that of the particular boundary conditions. Fortunately for the computer, all the problems used for illustration happened to possess linear boundary conditions. Nevertheless, this was not a limitation to the usefullness of the scheme, but it did reduce the amount of computation required for fixing the m constants. Also, most problems of practical importance have linear boundary conditions.

The most general linear boundary conditions could be represented by the matrix equation

$$B\ Y = K$$

where B is the coefficient matrix, Y is the column matrix with y, y^{i}, y^{ii}, y^{iii},---, $y^{(m-1)}$ for its elements, and K is the constant column.

$$y = \sum_{i=1}^{N_c} c_i x^{i-1}$$

$$D^p y = \sum_{i=p+1}^{N_c} \frac{(i-1)!}{(i-p-1)!}\, c_i x^{i-p-1}$$

By this time all but the first m coefficients c_i were known. Hence, at boundary points, where x is specified, y and any of its derivatives were just some linear combination of the c_i(i=1,2,---, m). Hence, the unknown column Y could be written as

$$Y = S\,C + T$$

where C is the column of undetermined coefficients c_i, and S and T were respectively the square and column matrices determined as:

$$s_{ij} = \begin{cases} \frac{(j-1)!}{(j-i)!}\, x^{j-1} & , j \geq i \\ 0 & , j < i \end{cases}$$

$$t_i = \frac{(j-1)!}{(j-i-1)!}\, x^{j-i-1}$$

After substitution

$$B\,(SC + T) = K$$

$$BSC = K - BT$$

and $$C = (BS)^{-1}\,(K - B\,T)$$

$$= DF$$

where $$D = (BS)^{-1} \qquad \text{(III-1)}$$

$$F = K - BT.$$

The subroutine DMTRX was programmed to find the D matrix while subroutine BOUND constructed the column vector F. Finally, the constants c_i were obtained by matrix multiplication.

The problem of obtaining corresponding matrices for obtaining modes corresponding to higher eigenvalues is discussed later (Intermediate Eigenvalues p. 34)

Subroutines

A variety of operations were expected to be performed repeatedly. Subroutines were programmed so that they could be inserted in any other programs and called for whenever desired. These subroutines along with their limitations are briefly discussed below.

i) Subroutine DIFFER (C,NC,D,ND,N): This subroutine was used to differentiate a given polynomial C having Nc terms, N times to obtain a polynomial D. Nd, the number of terms in D equaled Nc minus N. The only restriction on N was that it should be non-zero positive integer.

ii) Subroutine INTGRA (C,NC,D,ND,N): This subroutine gave a polynomial D as the Nth integration of the polynomial C. Nd equalled Nc plus N. N should be a non-zero positive integer. The constants of integrations would be left undetermined.

iii) Subroutine SEMUL (A,NA,B,NB,C,NC): The polynomials A and B were multiplied to get polynomial C by this subroutine. Na and Nb both should be non-zero positive integers.

iv) Subroutine SEDIV (A,NA,B,NB,C,NC): This subroutine was meant to obtain the polynomial C as the division of polynomial A by polynomial B. Nc, the number of terms in C, should be specified and the process is terminated after calculating that many terms.

v) Subroutine INVRS (A,N): This subroutine was designed to find the inverse of a square matrix A of order N by Gauss-Sidel reduction process. In case the inverse did not exist the program had instruction to write out 'NO INVERSE'.

vi) Subroutine MATMUL (A,N,B): This subroutine was specifically made to multiply a square matrix A, of order N, by a column matrix B.

vii) Subroutine INTGRD: The aim of the subroutine INTGRD was to construct $G[y^{(r)}]$ from the rth iterate. The sequence of instructions solely depended upon the operator M (and N, if present) of the specific problem. This subroutine, in case of eigenvalue problems, normalized the iterate every time and also computed the Raleigh's quotient, if necessary.

viii) Subroutine INITL: This subroutine inilized the particular problem.

ix) The main program: This program controlled the order in which all other subroutines were called.

x) Subroutine RESULT: This subroutine writes out what is supposed to be the answer.

CHAPTER IV

ILLUSTRATIVE PROBLEMS

1) Equilibrium Problems:

i) A Beam on Elastic Foundation:- The equilibrium of a flexible beam subjected to a uniformly distributed load, while resting on a continuous elastic foundation was considered. The non-dimensional formulation as taken from 'Engineering Analysis' by Crandall ([3], p. 195) was

$$y^{iv} + y = 1 \qquad 0 \leq x \leq 1$$

$$\text{and } y_{(0)} = y''_{(0)} = 0 \tag{IV-1}$$

$$y_{(1)} = y''_{(1)} = 0 \quad .$$

In problems of this type, which possess unique non-zero solutions, the iteration could be started from an initial guess

$$y^{(0)} = 0.$$

In the reference the problem was solved in several ways. The solution by Ritz's stationary functional method with trial family ([3], p. 235)

$$y = c_1 \, x \, (1-x) + c_2 \, x^2 \, (1-x^2)$$

was taken for comparison. The solution contained only five terms,

did not satisfy all the boundary conditions, and therefore is not exact. It is compared with the first, third and sixth iterates, in Table I.

Table I

Results: A Beam on Elastic Foundation

Coeff. Number	Comparison Solution	Iterations First	Third	Sixth
1	0.0000000	0.0000000	0.0000000	0.0000000
2	0.041249	0.041667	0.041249	0.041249
3	0.000032	0.000000	0.000000	0.000000
4	-0.082434	-0.083333	-0.082646	-0.082646
5	0.041217	0.041667	0.041667	0.041667
6			-0.000344	-0.000344
7			0.000000	0.000000
8			0.000098	0.000098
9			-0.000025	-0.000025
10			0.0000001	0.0000001

ii) Bending of a Strut:- This problem with its non-dimensional formulation was taken from "The Numerical Treatment of Differential Equations" by Collatz (p. 143). It considered the bending of a strut with varying flexural rigidity, and axial compressive load, by a distributed transverse load. The equations of equilibrium were

$$y^{ii} + (1 + \bar{x}^2)\, y + 1 = 0 \qquad -1 \leq \bar{x} \leq 1$$
$$y_{(-1)} = y_{(+1)} = 0. \tag{IV-2}$$

To reduce this formulation to the desired form, a change of variable was made.

$$2x = \bar{x} + 1. \tag{IV-3}$$

With this

$$y^{ii} + 4\,(1 + (2 - 4x^2)\, y) = 0 \qquad 0 \leq x \leq 1$$
$$y_{(0)} = y_{(1)} = 0. \tag{IV-4}$$

The first seven terms of the power series solution of the formulation (IV-2) were available in the reference (p. 225). For comparison, the change of variable (IV-3) was made. The solution appears in the second column of Table II.

It may be noted that the initial guess

$$y^{(0)} = 1$$

did not satisfy any of the boundary conditions. This did not in anyway prevent convergence, since at the very next opportunity the boundary conditions were forced on the iterate. The results are given in Table II.

Table II

Results: Bending of a Strut

Coeff. Number	Comparison Reference Solution	Iterations Second	Seventh	Twelfth
1	0.775×10^{-8}	0.00000	0.00000	0.00000
2	3.47293	4.66667	3.48506	3.47318
3	-1.99998	-6.00000	-2.00000	-2.00000
4	-4.63078	2.66667	-4.66578	-4.63129
5	+5.96531	-1.33333	5.99912	5.96463
6	-2.53309		-2.51655	-2.52592
7	-2.96677		-3.05473	-2.99460
8	4.43757		4.57156	4.51827
9	-1.80832		-1.99472	-1.99803
10	-0.96844		-0.65345	-0.60670
11	2.068897		1.55658	1.51371
12	-1.58388		-0.91128	-0.90380
13	0.731512		0.05860	0.07669
14	-0.21578		0.27760	0.26406
15	0.03083		-0.20495	-0.21599

2) Eigenvalue Problems:

i) Vibrations of a Beam Fixed at One End and Hinged at Another:- This problem, being very common, could be found in almost every text on vibration of elastic bodies. The present formulation appeared on page 255 of the reference [20].

$$\begin{aligned} &\bar{y}^{iv} = \beta^4 \bar{y} && 0 \leq \bar{x} \leq L \\ &\bar{y}_{(0)} = \bar{y}^{i}_{(0)} = 0 \\ &\bar{y}_{(L)} = \bar{y}^{ii}_{(L)} = 0 \end{aligned} \tag{IV-5}$$

To make it dimensionless, new variables were introduced-

$$\begin{aligned} y &= \bar{y}/L \\ x &= \bar{x}/L. \end{aligned} \tag{IV-6}$$

The new formulation was

$$\begin{aligned} &y^{iv} = \lambda\, y && 0 \leq x \leq 1 \\ &y_{(0)} = y^{i}_{(0)} = 0 \\ &y_{(1)} = y^{ii}_{(1)} = 0. \end{aligned} \tag{IV-7}$$

and $\lambda = \beta^4 L^4$.

The exact solution to this formulation is

$$= (\cos \alpha x - \cosh \alpha x) + \gamma(\sin \alpha x - \sinh \alpha x)$$

where $\alpha = \lambda^{\frac{1}{4}} = \beta L$

$$\gamma = \frac{\cos \alpha - \cosh \alpha}{\sinh \alpha - \sin \alpha}.$$

The α's are given by the equation

$$\text{Tan } \alpha = \text{Tanh } \alpha .$$

The exact solution corresponding to the smallest eigenvalue was expanded in a power series and the various iterates compared with it. (Table III).

Table III

Results: Beam Vibrations - First Mode

Coeff. Number	Comparison Solution	Iterations Tenth	Fifth	Second
3	1.0000000	1.0000000	1.0000000	1.0000000
4	-1.3098845	-1.3098847	-1.3098874	-1.3157893
7	0.6603361	0.6603360	0.6603518	0.7368418
8	-0.3706989	-0.3706989	-0.3707157	-0.5263155
9	0.0000000	0.0000000		0.1052631
11	0.0311460	0.0311460	0.0311551	
12	-0.0111266	-0.0111266	-0.0111328	
15	+0.0003082	0.0003082	0.0003095	
16	-0.0000807	-0.0000807	-0.0000814	
19	0.0000010	0.0000010	0.0000011	
20	-0.0000002	-0.0000002	-0.0000003	
λ_1		0.00420661	0.00420651	0.00376984

All $(4n+1)^{st}$ and $(4n+2)^{nd}$ coeff. are zeros except if last term of iteration.

Comparison solution is series expansion of

$$(\cos\beta_1 x - \cosh\beta_1 x) + \frac{\cos\beta_1 - \cosh\beta_1}{\sinh\lambda_1 - \sin\lambda_1}(\sin\beta_1 x - \sinh\beta_1 x)$$

with $\beta_1 = 3.9266023 = \lambda_1^{\frac{1}{4}}$, the eigenvalue obtained from the tenth iteration.

ii) Transverse Vibrations of a Tapered Beam of Rectangular Cross-section, Simply Supported:- This problem was studied by the author as an assignment for the course 'Machine Vibrations II'. The derivation of the formulation follows:

In addition to the usual assumption of small deflections, it was assumed that the taper is very small: so that the differential equation of motion can be approximated by that for a uniform beam (see formulation (IV-5)).

$$\bar{y}^{iv} = \beta^4 \bar{y} \qquad 0 \leq \bar{x} \leq L$$

with end conditions

$$\bar{y}_{(0)} = \bar{y}''_{(0)} = 0$$

$$\bar{y}_{(L)} = \bar{y}''_{(L)} = 0.$$

Only, the term $\beta^4 = \frac{\rho A}{EI}\omega^2$ ([20], p. 255) is no longer constant since A and I are variable for the problem considered here, b, width of the section was assumed to be uniform and the depth was assumed to vary linearly with $\bar{x}$.

$$d_{(x)} = d_1 + (d_2 - d_1)\frac{\bar{x}}{L}$$

$$= d_1\left[1 + (\alpha - 1)\frac{\bar{x}}{L}\right]$$

where d_1 and d_2 are depths at left and right ends and $\alpha = d_2/d_1$.

$$\frac{A_{(x)}}{I_{(x)}} = \frac{b\ d_{(2)}}{b\ d_{(x)}^3/12}$$

$$= \frac{12}{d_1^2\left[1+(\alpha-1)\frac{\bar{x}}{L}\right]^2}$$

$$\beta^4 = \frac{12\rho\omega^2}{Ed_1^2}\ \frac{1}{1+(\alpha-1)\frac{\bar{x}}{L}^2}$$

With this the differential equation becomes

$$1+(\alpha-1)\frac{\bar{x}}{L}^2\ \bar{y}^{iv} = \frac{12\rho\omega^2}{Ed_1^2}\ \bar{y}.$$

With the change of variable (IV-6)

$$[1+(\alpha-1)x]y^{iv} = \lambda y \qquad 0 \leq x \leq 1$$

$$y_{(0)} = y''_{(0)} = 0 \qquad \text{(IV-8)}$$

$$y_{(L)} = y''_{(L)} = 0.$$

where $\lambda = \frac{12\rho\omega^2}{Ed_1^2}$.

The finite difference method with ten intervals was used to solve the problem. For various values of α, the modes corresponding to the smallest eigenvalues were determined. These solutions formed the basis for comparison for the answers given by the present technique.

Good agreement was found for solutions with values of α = 1, 1.1,---,1.9. But for $\alpha = 2$ the process apparently converged to a solution which did not agree with that obtained by the finite difference method. An investigation into the matter revealed that the reason was in the divergence of the power series expansion of $1/(1+x)^2$. Forming this series was an important step in constructing the integrand at each iteration. As might be expected the error was corrected by turning the beam about its centre. The value of α was now 1/2 and the expansion of $1/(1-0.5x)^2$ converges rapidly enough. The various solutions are listed in Table IV, along with the solutions by the finite difference method.

It was expected that the analytical solution would be more accurate, since the difference method used only ten intervals, and inversion of a matrix of order nine was involved. This was easily verified for $\alpha = 1$, (see Table V) that is a uniform beam. For this case the mathematically exact solution was known. It was found that the answers from the difference method were a bit too low. There was no way to compare the accuracies of the two methods in case of other values of α. But one thing was certain, that the same trend (the solutions by the present technique being slightly larger than those by the difference method) was followed throughout.

Table IV

Results: Vibrations of Tapered Beam

Dia Ratio α	λ_{1f} Difference method Comparison Solution	λ_{1p} Polynomial exp.	$(\frac{\lambda_{1p}-\lambda_{1f}}{\lambda_{4p}})100$
1.0	3.1280996	3.1415925	0.41039377
1.1	3.2051857	3.2183947	0.41042200
1.2	3.2785025	3.2920185	0.41056877
1.3	3.3490291	3.3628388	0.41065602
1.4	3.4170429	3.4311626	0.41151387
1.5	3.4828436	3.4972445	0.41177847
1.6	3.5466312	3.5612992	0.41187216
1.7	3.6085762	3.6235085	0.41188344
1.8	3.6688213	3.6835085	0.39872854
1.9	3.7275184	3.7123093	-0.40969377
2.0	3.7847778	*3.8005053	0.41382655

*This value was obtained by turning the beam about its center and working the problem with $\alpha = 0.5$.

Table V

Results: Tapered Beam Problem with

$\alpha = 1$ i.e. a uniform beam.

Coeff. Number	Exact Solution Comparison Solution	Present Solution Tenth Iteration
λ_1	π	3.1415925
2	1.0000000	1.0000000
4	-1.6494792	-1.6449338
6	0.8162345	+0.8117423
8	-0.1923374	-0.1907517
10	0.0264380	+0.0261478
12	-0.0023787	-0.0023460
14	+0.0001509	0.0001484
16	-0.0000071	0.0000070
18	0.00000025	0.00000025
20	Less than 10^{-8}	Less than 10^{-8}

Note:- All odd coefficients in both solutions are zeros.

iii) Longitudinal Vibrations of a Cantilever of Varying Cross-Section:- This eigenvalue problem was formulated and solved in reference ([2], p. 147). The formulation given was

$$(1+x)y^{ii} + y^{i} + \lambda (1+x)y = 0 \qquad 0 \leq x \leq 1$$
$$y^{i}_{(0)} = y_{(1)} = 0. \tag{IV-9}$$

Various numerical methods were applied and close bounds for the smallest eigenvalue were given (2 , p. 236).

$$3.218211 \leq \lambda_1 \leq 3.218532$$

In order to compare this solution with the solution obtained by the present method, a solution was attempted by another method. The various steps involved were: 1) The first m terms and the eigenvalue were assumed. 2) About 40 more terms in the series expansion were found by using the recurrence relation obtained from the differential equation. 3) The first m terms were then readjusted to satisfy the boundary conditions. 4) An improved eigenvalue was calculated from the Raleigh's quotient of the polynomial solution. The cycle consisting of steps two to four was repeated a number of times, everytime improving the solution function and the eigenvalue.

However, it was observed that the convergence was very slow in both of these processes. The coefficients in the solution polynomial did not decrease rapidly enough. And even after about 20 iterations the eigenvalue agreed with the correct one, only to the second significant figure. To accelerate the convergence, a variable change was made-

$$t = 2x$$

so that the new formulation was

$$4(1+1/2t)\frac{d^2y}{dt^2} + 2\frac{dy}{dt} + \lambda(1+1/2t)y = 0$$

$$0 \leq t \leq 2$$

$$y'_{(0)} = y_{(2)} = 0.$$

With this formulation agreement up to five places was obtained in 19 iterations. More iterations deteriorated the solution rather than improving. The reason could be attributed to the fact that only eight place arithemetic was used, and this might perhaps be the closest, the answer can reach the true solution with the number of places used in the arithemetic. The solutions are compared in table VI.

Table VI

Results: Longitudinal Vibrations of a Cantilever

Coeff. Number	Comparison Solution	Iterations 6th	13th	19th
λ	3.2184796	3.2427394	3.2245237	3.2185929
1	1.0000000	1.0000000	1.0000000	1.0000000
2	0.0000000	0.0000000	0.0000000	0.0000000
3	-0.3966840	-0.3826942	-0.3909474	-0.3940352
4	0.0661140	0.0598456	0.0630494	0.0642405
5	0.0018139	0.0023530	0.0020571	0.0019467
6	0.0045958	0.0040887	0.0043546	0.0044540
7	-0.0028505	-0.0024963	-0.0026736	-0.0027404
8	0.0011162	0.0009618	0.0010388	0.0010680
9	-0.0004800	-0.0004102	-0.0004452	-0.0004585
10	0.0002170	0.0001836	0.0002002	0.0002064
11	-0.0000983	-0.0000824	-0.0000903	-0.0000933
12	0.0000445	0.0000373	0.0000410	0.0000425
13	-0.0000206	-0.0000170	-0.0000188	-0.0000195
14	0.0000095	0.0000078	0.0000087	0.0000090
15	-0.0000044	-0.0000033	-0.0000040	-0.0000042
16	0.0000021		0.0000019	0.0000019
17	-0.0000010		-0.0000009	-0.0000009
18	0.0000005		0.0000004	0.0000004
19	-0.0000002		-0.0000002	-0.0000002
20	0.0000001		0.0000001	0.0000001

iv) Intermediate Eigenvalues:- The basic iteration procedure always converged to the smallest eigenvalue and the corresponding eigenfunction. However, the method could be modified to provide other modes by orthogonalising the initial trial with respect to the known modes and continually purifying the iterates.

Thus the iterate was required to satisfy all the boundary conditions and the orthogonality conditions, a total of (m+p) conditions, where the (p+1)st mode was sought. This necessiated (m+p) undetermined terms in the expansion polynomial. Out of these, m terms were supplied by the constants of integration, and the next p terms in the series were used to provide the rest.

The orthogonality condition, in the integral equation form, cannot be directly used. The conversion to algebraic equation form is illustrated here. Even though y_1 and y_2, the first and the second eigenfunctions, are used in the illustration, the process is perfectly general. The condition for y_1 and y_2 may be written as

$$\int_0^1 y_2 M[y_1]\, dx = 0$$

or
$$\int_0^1 y_2 N[y_1]\, dx = 0. \qquad \text{(IV-10)}$$

Both the forms are equivalent and simplicity will govern the choice. For the sake of illustration the later form is chosen.

The first eigenfunction, y_1, would be known in polynomial form. The application of the operator N will yield another polynomial

$$N[y_1] = \sum_i b_i x^{i-1}.$$

y_2 is sought as a power series

$$y_2 = \sum_j c_j x^{j-1}.$$

Substituting in (IV-10)

$$\int_0^1 y_2 N[y_1]\,dx = \int_0^1 (\sum_j c_j x^{j-1})(\sum_i b_i x^{i-1})\,dx$$

$$= \sum_j c_j \int_0^1 \sum_i (b_i x^{i+j-2})\,dx$$

$$= \sum_j c_j \sum_i \frac{b_i}{i+j-1} (x^{i+j-1})_0^1$$

$$= \sum_j c_j \sum_i \frac{b_i}{i+j-1}$$

$$= \sum_j c_j d_j = 0$$

where $d_j = \sum_i \frac{b_i}{i+j-1}$.

Thus, the conditions the iterate should satisfy, are

$$B_i[y] = 0 \qquad i = 1, 2, \ldots\ldots, m$$

$$\sum_j c_j d_j = 0 \qquad i = 1, 2, \ldots\ldots, p.$$

The method was applied to find the eigenfunction corresponding to the second lowest eigenvalue of the formulation (IV-1). The result, with the expansion of true solution, is tabulated in Table VII.

Table VII

Results: Beam Vibrations - Second Mode

Coeff. Number	Comparison Solution	Iterations Tenth	Fifth	Second
$1/\lambda_2$		0.00040057	0.00040205	0.000509232
3	1.0000000	1.0000000	1.0000000	1.0000000
4	-2.3561851	-2.3561802	-2.3550217	-2.2766555
7	6.9345492	6.9345261	6.9090413	5.4548292
8	-7.0024633	-7.0024360	-6.9620717	-4.8558406
11	3.4348552	3.4348092	3.3694359	1.1291510
12	-2.2072238	-2.2071817	-2.1463823	-0.4514840
15	0.3569306	0.3509095	0.3287323	
16	-0.1681989	-0.1681846	-0.1496814	
19	0.0121331	0.0121300	0.0087900	
20	-0.0045139	-0.0045122	-0.0028828	
23	0.0001725	0.0001723	0.0000522	
24	-0.0000530	-0.0000529	-0.0000100	
27	0.0000012	0.0000012		
28	-0.0000003	-0.0000003		

All $(4n+1)^{st}$ and $(4n+2)^{nd}$ coeff. are zeros.

Comparison solution =

$$(\cos\beta_2 x - \cosh\beta_2 x) + \frac{\cos\beta_2 - \cosh\beta_2}{\sinh\beta_2 - \sin\beta_2} (\sin\beta_2 x - \sinh\beta_2 x)$$

with $\beta_2 = \lambda_2^{\frac{1}{4}} = 7.0685465$

obtained from 10th iteration.

CHAPTER V

THE FINAL PROBLEM

Most of the problems considered up to this time were of academic nature. It was now decided to work a problem of practical interest. The problem selected is associated with the name of Graetz [7], who obtained the first analytical solution of the problem. The present non-dimensional formulation was taken from a Master's Thesis by Robert Lipkis [10].

$$y^{ii} + y^{i}/x + (1-x^2)\, y = 0 \qquad 0 \leq x \leq 1$$

$$y^{i}_{(0)} = y_{(1)} = 0 \tag{V-1}$$

These equations represent the problem of 'Heat Transfer to an Incompressible Fluid in Laminar Motion'. A brief description of the problem is given in Appendix C.

Mr. Lipkis had evaluated the first few eigenvalues to a very good accuracy. The present method was applied only to obtain the first mode. The eigenvalue obtained (7.31358) tallys with that given in the reference (7.31358) to the given number of places.

An interesting thing about this problem is that, the quotient

$$\int_0^1 M\, y\, dx \Big/ \int_0^1 N\, y\, dx$$

for an admissible function y is also a good approximation to the

eigenvalue. In fact, when this quotient was used by mistake, the iteration converged a little earlier than it did while using Raleigh's quotient. This solution is listed in column 3 of the Table VIII.

The reference [17] also listed the values of y^1 at x = 1. The value of $y^1_{(1)}$, calculated from the polynomial solution by the present method compared very well with the value in reference. However, this does not mean much. Because, the mode function is only determined to within a constant multiple which can always be adjusted for any desired value of $y^1_{(1)}$. It only means that both solutions were normalized in some equivalent manner.

A solution was also attempted without the use of Raleigh's quotient. The term factored out during normalization was taken to be the reciprocal of an approximation to eigenvalue. The process diverged, oscillating between two values which bracketed the true value. As a remedy a weighted average of two successive iterates was tried. This effected the convergence. But the convergence was not fast enough. It is quite likely that a proper choice of weighting factors would do better. (The weighting procedure used was nth approximation = 0.7 times the (n-1)st approximation + 0.3 times the nth iterate.) This solution appears in the second column of Table VIII, and the solution by iteration using Raleigh's quotient appears in column 4.

Table VIII

Results: Heat Transfer to an Incompressible Fluid in Laminar Motion

Coeff. Number	Comparison Solution 1	Comparison Solution 2	Tenth Iteration
$1/\lambda_1$	0.13677533	0.13673189	0.13073178
1	1.0000000	1.0000000	1.0000000
3	-1.8366730	-1.8283964	-1.8283971
5	1.3018585	1.2928571	1.2928585
7	-0.6285497	-0.6340986	-0.6340993
9	0.2080135	0.2202023	0.2202023
11	-0.0536991	-0.0624800	-0.0624797
13	0.0105566	0.0143570	0.0143570
15	-0.0017016	-0.0028671	-0.0028671
17	0.0002170	0.0004921	0.0004921
19	-0.0000236	-0.0000758	-0.0000758
21	0.0000020	0.0000104	0.0000104
23	-0.0000002	-0.0000013	-0.0000013
25	0.0000000	0.0000001	0.0000001

CHAPTER VI

CONCLUSION AND DISCUSSION

Even though a rigourous proof of the convergence of the iteration for a general problem is not given, it could be seen that the process converges in many cases. The method is sufficiently general in application, and could be used with advantage when no other simple analytical solution is available.

The method is equally applicable to equations with constant and non-constant coefficients. The only requirement is that the coefficients should possess a power series expansion. Obviously, only a finite number of terms (usually a very few) could be kept, so that the error involved will be dependent on the number of terms used and therefore on the rate of convergence of the series expansions of the coefficients.

To be able to continue the iteration without getting stuck in the integration process, no negative or fractional powers of the independent variable should appear in the result of $G[y]$. The appearance of negative powers is usually due to some coefficient in the differential equation, when solved for the highest derivative term. This difficulty can, in most cases, be overcome by a suitable change of variable e.g. $a + t = x$, $|a| > 1$.

Another difficulty is encountered when some coefficient has $(a \pm x)$ in the denominator with, $|a|$ less than, equal to or slightly larger than one. Division by $(a \pm x)$ is equivalent to

multiplication by its reciprocal. The coefficients in the expansion of $1/(a \pm x)$, decrease, remain the same viz. unity, or increase according as $|a|$ is greater than, equal to, or less than unity. It is of utmost importance to see that these coefficients decrease sufficiently rapidly, by effecting a change of variable, if necessary. Otherwise, the chances of convergence go down.

This case was encountered in the problem of vibrations of a tapered beam with $\alpha = 2$, and in the longitudinal vibrations of a cantilever. In the first case, the difficulty was solved by switching the ends of the beam and thus working with $\alpha = 1/2$. This in effect is equivalent to a change of variable of $1 - t = x$. In the latter problem a variable change of $t = 2x$, was made. Both of these changes sufficiently accelerated the convergence.

In the end, a few words about the use of various subroutines would not be out of place. Except for subroutines INTGRD and INITL, all subroutines are very general in nature, and can be used for almost any problem. However, when the order of the equation is small e.g. 2, the formation of the D matrix is a very simple matter. If it is fed to the program as data, considerable computer core-space will be saved. This can be used, with advantage, for carrying out more iterations, or for employing more places of arithmetic. The same comment applies to the use of the subroutine BOUND. For instance, the problem of longitudinal vibrations of a cantilever had the following boundary conditions-

$$y^{i}_{(0)} = y_{(1)} = 0$$

The condition $y^{i}_{(0)} = 0$ leads to $c_2 = 0$, and then $y_{(1)} = 0$, means simply

$$c_1 = \sum_{i=3} c_i .$$

If this is recognized in advance, a few instructions can replace the subroutines BOUND and MATMUL (A,N,B), and therefore subroutines DMTRX and INVRS (A,N).

SELECTED
BIBLIOGRAPHY

1. Bennett, Albert A, William E. Milne, and Harry Bateman. Numerical Integration of Differential Equations. pp. 51-70. New York: Dover Publications, Inc., 1956.

2. Collatz, Lothar. The Numerical Treatment of Differential Equations. Third Edition.

   Translated by P. G. Williams from a supplemented version of the second German edition. Springer-Verlog. Ohg. Berlin.Gottingten.Heidelberg, 1960.

3. Crandall, Stephen H. Engineering Analysis-A Survey of Numerical Procedures. New York: McGraw-Hill Book Co., Inc., 1956.

4. Den Hartog, J. P. Advanced Strength of Materials. New York: McGraw-Hill Book Company, Inc., 1952.

5. Fox, L. The Numerical Solution of Two-point Boundary Problems in Ordinary Differential Equations. London: Oxford University Press, 1957.

6. Hilderbrand, Francis B. Advanced Calculus for Applications. pp. 200-216. New Jersey: Prentice-Hall, Inc., 1964.

7. Jacob, M. Heat Transfer. Vol. I. p. 451. New York: John Wiley & Sons, Inc., 1949.

8. Levy, H., and E. A. Bagget. Numerical Solutions of Differential Equations. Vol. I. First American Edition. pp. 35-62. New York: Dover Publications, Inc., 1950.

9. Lipkis, Robert P. 'Discussion of Paper by Sellars, Tribus, and Klein' ASME Transactions. 1956, 78:1272-1274.

10. Lipkis, Robert P. Heat Transfer to an Incompressible Fluid in Lanienar Motion. A Master's Thesis, University of California, Los Angelas, California, August, 1954.

11. Lovitt, William V. Linear Integral Equations. First Edition. New York: McGraw-Hill Book Company, Inc., 1924.

12. Milne, William E. Numerical Solution of Differential Equations. pp. 37-42. New York: John Wiley & Sons, Inc., 1953.

13. Picard, E. Traite d'analyse. Vol. III chapter VI, Paris: Gauthier Villars & Cie, 1896.

14. Salvadori, Mario G., and Melvin L. Baron. Numerical Methods in Engineering. New York: Prentice-Hall, Inc., 1952.

15. Scarborough, James B. Numerical Mathematical Analysis. pp. 218-260. Baltimore: The John Hopkins Press, 1950.

16. Schwarz, H. A. Gosamnette Mathematische Aphandlringen von H. A. Schwarz. Vol. I. pp. 241-265. Berlin, 1890.

17. Siegel, Robert. 'Heat Transfer for Laminar Flow in Ducts With Arbitrary Time Variations in Wall Temperature.' ASME Transactions. 1960, 82(3):241-249.

18. Stodola, Aurel. Steam and Gas Turbines. Vol. I. Translated by Louis C. Loewenstein from the Sixth German Edition. pp. 460-465. New York: McGraw-Hill Book Company, Inc., 1927.

19. Tien, Hsun. Heat Transfer for Incompressible Fluid in Laminar Motion in Circular Tubes. Master's Report. Kansas State University, Manhattan, Kansas, 1959.

20. Tong, Kin L. Theory of Mechanical Vibrations. New York: John Wiley & Sons, Inc., 1960.

21. Vianello, L. 'Graphische Untersuchring der Knickfestigkeit gerader stabe'. Z. Ver. deut. Ing., 1898, 42:1436-1443.

APPENDIX A

Computer Programs

The programs for the problems illustrated in this work are listed here. Some of the subroutines, which are of rather general character are listed in Appendix B. And Appendix C gives brief description of the Final Problem, and the computer program in its entirety.

```
      SUBROUTINE INITL
C     THIS SUBROUTINE INITIALIZES THE PROBLEM OF
C     A BEAM ON ELASTIC FOUNDATION
      COMMON M,N,IT,NC,NE,NP,MP1,NP1,EI
      COMMON B(8,8),D(8,16),C(100),E(10),P(100)
      C(1)=1.
      NC=1
      CALL BMTRX
      RETURN
      END
```

```
      SUBROUTINE INTGRD
C     THIS SUBROUTINE FORMS
C     THE INTIGRAND FOR THE PROBLEM OF
C     A BEAM ON ELASTIC FOUNDATION
      COMMON M,N,IT,NC,NE,NP,MP1,NP1,EI
      COMMON B(8,8),D(8,16),C(100),E(10),P(100)
    1 FORMAT(4E16.8)
    2 FORMAT(/)
      WRITE(3,1) (C(I),I=1,NC)
      WRITE(3,2)
      C(1)=1.-C(1)
      DO 20 I=2,NC
   20 C(I)=-C(I)
      WRITE(3,1) (C(I),I=1,NC)
      RETURN
      END
```

```
      SUBROUTINE INITL
C     THIS SUBROUTINE INITIALIZES THE PROBLEM OF
C     BENDING OF A STRUT
      COMMON M,N,IT,NC,NE,NP,MP1,NP1,EI
      COMMON B(8,8),D(8,16),C(100),E(10),P(100)
      E(1)=2.
      E(2)=-4.
      E(3)=4.
      NE=3
      CALL DMTRX
      C(1)=1.
      NC=1
      RETURN
      END
```

```
      SUBROUTINE INTGRC
C     THIS SUBROUTINE FORMS
C     THE INTIGRAND FOR THE PROBLEM OF
C     BENDING OF A STRUT
      COMMON M,N,IT,NC,NE,NP,MP1,NP1,EI
      COMMON B(9,8),D(8,16),C(100),E(10),P(100)
    1 FORMAT(4E16.8)
    2 FORMAT(/)
      WRITE(3,1) (C(I),I=1,NC)
      WRITE(3,2)
      CALL SCMUL (C,NC,E,NE,P,NP)
      WRITE(3,2)
      C(1)=-4.-4.*P(1)
      DO 20 I=2,NP
   20 C(I)=-4.*P(I)
      NC=NP
      WRITE(3,1) (C(I),I=1,NC)
      RETURN
      END
```

```
      SUBROUTINE INITL
C     THIS SUBROUTINE INITIALIZES THE PROBLEM OF
C     VIBRATIONS OF A BEAM FIXED AT ONE END
C     AND HINGED AT ANOTHER - FIRST MODE
      COMMON M,N,IT,NC,NE,NP,NP1,NP1,EI
      COMMON B(8,8),D(8,16),C(100),E(10),P(100)
      C(1)=1.
      NC=1
      CALL DMTRX
      RETURN
      END
```

```
      SUBROUTINE INTGRD
C     THIS SUBROUTINE NORMALIZES THE ITERATE FOR THE PROBLEM -
C     VIBRATIONS OF A BEAM FIXED AT ONE END
C     AND HINGED AT ANOTHER - ANY MODE
      COMMON M,N,IT,NC,NE,NP,MP1,NP1,EI
      COMMON B(8,8),D(8,16),C(100),E(10),P(100)
    1 FORMAT(4E16.8)
    2 FORMAT(/)
      DO 20 I=1,NC
      IF(C(I).NE.0.0) GO TO 40
   20 CONTINUE
   40 EI=C(I)
      WRITE(3,1) EI
      WRITE(3,2)
      DO 60 J=I,NC
   60 C(J)=C(J)/EI
      WRITE(3,1) (C(I),I=1,NC)
      RETURN
      END
```

```
      SUBROUTINE INITL
C     THIS SUBROUTINE INITIALIZES THE PROBLEM OF
C     VIBRATIONS OF A TAPERED BEAM OF RECTANGULAR
C     CROSS-SECTION, SIMPLY SUPPORTED
      COMMON M,N,IT,NC,NE,NP,MP1,NP1,EI
      COMMON B(8,8),D(9,16),C(100),E(10),P(100)
    2 FORMAT(14F5.1)
      CALL ONTRX
      READ(1,2) ALPH
      E(1)=1.
      E(2)=2.*(ALPH-1.)
      E(3)=(ALPH-1.)*(ALPH-1.)
      NE=3
      C(1)=1.
      RETURN
      END
```

```
      SUBROUTINE INTGRD
C     THIS SUBROUTINE NORMALIZES THE ITERATE,
C     AND FORMS THE INTEGRAND FOR THE PROBLEM OF
C     VIBRATIONS OF A TAPERED BEAM OF RECTANGULAR
C     CROSS-SECTION, SIMPLY SUPPORTED
      COMMON M,N,IT,NC,NE,NP,MP1,NP1,EI
      COMMON B(8,8),D(8,16),C(100),E(10),P(100)
    1 FORMAT(4E16.8)
    2 FORMAT(/)
      NC=40
C     NORMALIZATION OF THE ITERATE
      DO 10 I=1,NC
      IF(C(I).NE.0.0) GO TO 20
   10 CONTINUE
   20 EI=C(I)
      DO 40 J=I,NC
   40 C(J)=C(J)/EI
      WRITE(3,1)EI
      WRITE(3,2)
      WRITE(3,1) (C(I),I=1,NC)
C     CONSTRUCTION OF THE INTEGRAND
      NP=NC
      CALL SEDIV(C,NC,E,NE,P,NP)
      WRITE(3,2)
      DO 50 I=1,NC
   50 C(I)=P(I)
      WRITE(3,1) (C(I),I=1,NC)
      RETURN
      END
```

```
      SUBROUTINE INITL
C     THIS SUBROUTINE INITIALIZES THE PROBLEM OF
C     VIBRATIONS OF A CANTILEVER BEAM OF VARYING CROSS-SECTION
      COMMON M,N,IT,NC,NE,NP,MP1,NP1,EI
      COMMON B(8,8),D(8,16),C(100),E(10),P(100)
      E(1)=2.
      E(2)=1.
      NC=2
      CALL CNTRX
      C(1)=-4.
      C(2)=0.
      C(3)=1.
      NC=3
      RETURN
      END
```

```
      SUBROUTINE DMTRX
C     THIS SUBROUTINE CONSTRUCTS D MATRIX FROM B MATRIX
C     FOR THE PROBLEM OF
C     VIBRATIONS OF A CANTILEVER BEAM OF VARYING CROSS-SECTION
      DIMENSION F(10)
      COMMON M,N,IT,NC,NE,NP,MP1,NP1,EI
      COMMON B(8,8),D(8,16),C(100),E(10),P(100)
    1 FORMAT(4E16.8)
    2 FORMAT(14F5.2)
      F(1)=1.
      DO 20 I=2,M
   20 F(I)=F(I-1)*FLOAT(I-1)
      READ(1,2) ((B(I,J),J=1,M),I=1,M)
      DO 100 I=1,M
      IF(I.GT.N) GO TO 60
      DO 40 J=1,M
   40 D(I,J)=B(I,J)*F(J)
      GO TO 100
   60 D(I,1)=B(I,1)
      DO 80 J=2,M
      D(I,J)=B(I,1)
      DO 80 K=2,J
      JK=J-K
   80 D(I,J)=D(I,J)+F(J)*B(I,K)/F(JK+1)
  100 CONTINUE
      DO 120 I=1,M
      CON=1.
      DO 120 J=1,M
      D(I,J)=D(I,J)*CON
  120 CON=CON*2.
      CALL INVRS(D,M)
      WRITE(3,1) ((D(I,J),J=1,M),I=1,M)
      RETURN
      END
```

```
      SUBROUTINE INTGRD
C     THIS SUBROUTINE NORMALIZES THE ITERATE,
C     FINDS ITS RALEIGHS QUOTIENT AND FORMS
C     THE INTIGRAND FOR THE PROBLEM OF
C     VIBRATIONS OF A CANTILEVER BEAM OF VARYING CROSS-SECTION
      DIMENSION F(100),G(100),R(100)
      COMMON M,N,IT,NC,NE,NP,NP1,NP1,EI
      COMMON B(8,8),D(8,16),C(100),E(10),P(100)
    1 FORMAT(4E16.8)
    4 FORMAT(/)
C     NORMALIZATION OF THE ITERATE
      DO 10 I=1,NC
      IF(C(I).NE.0.0) GO TO 20
   10 CONTINUE
   20 EI=C(I)
      DO 40 J=I,NC
   40 C(J)=C(J)/EI
      WRITE(3,1)EI
      WRITE(3,4)
      WRITE(3,1) (C(I),I=1,NC)
C     CALCULATION OF RALEIGHS QUOTIENT
      CALL DIFFER (C,NC,F,NF,1)
      WRITE(3,4)
      CALL DIFFER (F,NF,G,NG,1)
      WRITE(3,4)
      CALL SEMUL (G,NG,E,NE,R,NR)
      WRITE(3,4)
      DO 50 I=1,NF
   50 R(I)=P(I)+F(I)/2.
      CALL SEMUL (C,NC,R,NR,G,NG)
      WRITE(3,4)
      CALL INTGRA (G,NG,R,NR,1)
      WRITE(3,4)
      DM =0.
      CON=1.
      DO 60 I=3,NR
```

```
      CON=CON*2.
   60 DM=DM+CON*R(I-1)
      CALL SEMUL (C,NC,E,NE,G,NG)
      WRITE(3,4)
      CALL SEMUL (C,NC,G,NG,R,NR)
      WRITE(3,4)
      CALL INTGRA (R,NR,G,NG,1)
      WRITE(3,4)
      DN=0.
      CON=1.
      DO 70 I=3,NG
      CON=CON*2.
   70 DN=DN-CON*G(I-1)
      EI=4.*DM/DN
      WRITE(3,1) EI,DM,DN
      WRITE(3,4)
C     CONSTRUCTION OF THE INTEGRAND
      CALL SEDIV (F,NF,E,NE,G,NC)
      WRITE(3,4)
      DO 80 I=1,NC
   80 C(I)=-EI*C(I)/4.-G(I)/2.
      WRITE(3,1) (C(I),I=1,NC)
      WRITE(3,4)
      RETURN
      END
```

```
      SUBROUTINE BOUND
C     THIS SUBROUTINE COMPUTES THE F VECTOR FOR THE PROBLEM OF
C     VIBRATIONS OF A CANTILEVER BEAM OF VARYING CROSS-SECTION
      DIMENSION Y(10)
      COMMON M,N,IT,NC,NE,NP,MP1,NP1,EI
      COMMON B(8,8),D(8,16),C(100),E(10),P(100)
    1 FORMAT(4E16.8)
    2 FORMAT(/)
      DO 10 I=1,M
   10 C(I)=0.
      CON=2.**M
      DO 30 I=1,M
      Y(I)=0.
      AI=I
      DO 20 J=MP1,NC
      Y(I)=Y(I)+CON*C(J)
   20 CON=CON*(FLOAT(J)/(FLOAT(J)+1.-AI))*2.
      CON=2.**M
      DO 30 J=1,I
   30 CON=CON*FLOAT(M+1-J)/2.
      WRITE(3,1) (C(I),I=1,M)
      RETURN
      END
```

```
C     MAIN PROGRAM FORR OBTAINING THE SECOND
C     EIGENFUNCTION FOR THE PROBLEM OF
C     VIBRATIONS OF A BEAM FIXED AT ONE END
C     AND HINGED AT ANOTHER
      DIMENSION R(100)
      COMMON M,N,IT,NC,NE,NP,MP1,NP1,EI
      COMMON B(8,8),D(8,16),C(100),E(10),P(100)
    1 FORMAT(4E16.8)
    2 FORMAT(1415)
    3 FORMAT(1H1)
    4 FORMAT(/)
      READ(1,2) M,N,ITL
      MP1=M+1
      NP1=N+1
      CALL INITL
      WRITE(3,3)
      DO 40 IT=1,ITL
      CALL INTGRD
      CALL INTGRA(C,NC,R,NR,M)
      DO 20 I=MP1,NC
   20 C(I)=R(I)
      CALL BOUND
      WRITE(3,4)
      CALL MATMUL(D,MP1,C)
   40 WRITE(3,3)
      CALL RESULT
      STOP
      END
```

```
      SUBROUTINE INITL
C     THIS SUBROUTINE INITIALIZES THE PROBLEM OF
C     VIBRATIONS OF A BEAM FIXED AT ONE END
C     AND HINGED AT ANOTHER - SECOND MODE
      DIMENSION R(100)
      COMMON M,N,IT,NC,NE,NP,MP1,NP1,EI
      COMMON D(8,8),B(8,16),C(100),E(10),P(100)
    1 FORMAT(4E16.8)
    2 FORMAT(14I5)
      READ(1,2) NP
      READ(1,1)(R(I),I=1,NP)
      DO 20 I=1,NP
      AI=I
      P(I)=0.
      DO 20 J=1,NP
   20 P(I)=P(I)+R(J)/(AI+FLOAT(J)-1.)
      WRITE(3,1) (P(I),I=1,NP)
      CALL CNTRX
      C(8)=1.
      NC=8
      CALL BOUND
      CALL MATMUL(D,MP1,C)
      C(7)=C(5)
      C(5)=0.
      RETURN
      END
```

```
      SUBROUTINE DMTRX
C     THIS SUBROUTINE CONSTRUCTS D MATRIX FOR THE PROBLEM OF
C     VIBRATIONS OF A BEAM FIXED AT ONE END
C     AND HINGED AT ANOTHER - SECOND MODE
      DIMENSION F(10)
      COMMON M,N,IT,NC,NE,NP,MP1,NP1,EI
      COMMON B(8,8),D(8,16),C(100),E(10),P(100)
    1 FORMAT(4E16.8)
    2 FORMAT(14F5.2)
    3 FORMAT(/)
      F(1)=1.
      DO 20 I=1,MP1
   20 F(I)=F(I-1)*FLOAT(I-1)
      READ(1,2) ((B(I,J),J=1,M),I=1,M)
      DO 100 I=1,M
      IF(I.GT.M) GO TO 60
      DO 40 J=1,MP1
   40 D(I,J)=B(I,J)*F(J)
      GO TO 100
   60 D(I,1)=B(I,1)
      DO 80 J=2,MP1
      D(I,J)=B(I,1)
      DO 80 K=2,J
      JK=J-K
   80 D(I,J)=D(I,J)+F(J)*B(I,K)/F(JK+1)
  100 CONTINUE
      DO 120 I=1,MP1
  120 D(MP1,I)=P(I)
      WRITE(3,1) ((D(I,J),J=1,MP1),I=1,MP1)
      CALL INVRS(D,MP1)
      WRITE(3,3)
      WRITE(3,1) ((D(I,J),J=1,MP1),I=1,MP1)
      RETURN
      END
```

```
      SUBROUTINE ROUND
C     THIS SUBROUTINE COMPUTES THE F VECTOR
C     IN THE MATRIX EQUATION CC=F FOR THE PROBLEM OF
C     VIBRATIONS OF A BEAM FIXED AT ONE END
C     AND HINGED AT ANOTHER - SECOND MODE
      DIMENSION Y(10)
      COMMON M,N,IT,NC,NE,NP,MP1,NP1,EI
      COMMON B(8,8),D(8,16),C(100),E(10),P(100)
    1 FORMAT(4E16.8)
      DO 20 I=1,MP1
   20 C(I)=0.
        CON=1.
      DO 60 I=1,M
      Y(I)=0.
      AI=I
      DO 40 J=8,NC
      Y(I)=Y(I)+CON*C(J)
   40 CON=CON*(FLOAT(J)/(FLOAT(J)+1.-AI))
      CON=1.
      DO 60 J=1,I
   60 CON=CON*(8.-FLOAT(J))
      DO 80 J=1,M
   80 C(I)=C(I)-B(I,J)*Y(J)
      DO 100 I=8,NC
  100 C(MP1)=C(MP1)-P(I)*C(I)
      WRITE(3,1) (C(I),I=1,MP1)
      WRITE(3,1) (C(I),I=1,NC)
      RETURN
      END
```

```
      SUBROUTINE INITL
C     THIS SUBROUTINE INITIALIZES THE PROBLEM OF
C     HEAT TRANSFER TO AN INCOMPRESSIBLE FLUID
C     IN LAMINAR MOTION
      COMMON M,N,IT,NC,NE,NP,MP1,NP1,EI
      COMMON B(8,8),D(8,16),C(100),E(10),P(100)
      CALL CMTRX
      E(1)=-1.
      E(2)=0.
      E(3)=1.
      NE=3
      C(1)=1.
      C(2)=0.
      C(3)=-1.
      NC=3
      RETURN
      END
```

```
      SUBROUTINE INTGRD
C     THIS SUBROUTINE NORMALIZES THE ITERATE,
C     FINDS ITS RALEIGHS QUOTIENT AND FORMS
C     THE INTIGRAND FOR THE PROBLEM OF
C     HEAT TRANSFER TO AN INCOMPRESSIBLE FLUID
C     IN LAMINAR MOTION
      DIMENSION F(100),G(100),H(100),R(100)
      COMMON M,N,IT,NC,NE,NP,NP1,NP1,EI
      COMMON B(8,8),C(8,16),C(100),E(10),P(100)
    1 FORMAT(4E16.8)
    2 FORMAT(/)
C     NORMALIZATION OF THE ITERATE
      DO 20 I=1,NC
      IF(C(I).NE.0.0) GO TO 30
   20 CONTINUE
   30 CIJ=C(I)
      DO 40 J=1,NC
   40 C(J)=C(J)/CIJ
      WRITE(3,1) (C(I),I=1,NC)
      WRITE(3,2)
C     CALCULATION OF RALEIGHS QUOTIENT
      CALL SEMUL(C,NC,E,NE,G,NG)
      WRITE(3,2)
      CALL SEMUL(C,NC,G,NG,H,NH)
      WRITE(3,2)
      F(1)=0.
      CALL INTGRA(H,NH,F,NF,1)
      WRITE(3,2)
      DN=0.
      DO 80 I=1,NF
   80 DN=DN+F(I)
      CALL DIFFER(C,NC,F,NF,1)
      WRITE(3,2)
      CALL DIFFER(F,NF,H,NH,1)
      WRITE(3,2)
      DO 60 I=1,NH
```

```
   60 H(I)=H(I)+F(I+1)
      WRITE(3,1) (H(I),I=1,NH)
      WRITE(3,2)
      CALL SEMUL (C,NC,H,NH,R,NR)
      WRITE(3,2)
      H(1)=0.
      CALL INTGRA (R,NR,H,NH,1)
      WRITE(3,2)
      DM=0.
      DO 70 I=1,NH
   70 DM=DM+H(I)
      H(1)=0.
      EI=DM/DN
      WRITE(3,1) EI,DM,DN
      WRITE(3,2)
C     CONSTRUCTION OF THE INTEGRAND
      NC=NG
      DO 90 I=1,NC
      C(I)=G(I)-F(I+1)/EI
      IF(I.GT.NF)  C(I)=G(I)
   90 CONTINUE
      R(1)=0.
      WRITE(3,1) (C(I),I=1,NC)
      WRITE(3,2)
      RETURN
```

APPENDIX B

Programs and Subroutines of General Nature

The main program and all the subroutines listed in this appendix are very general in nature, except that the subroutines DMTRX and BOUND can only be used if the boundaries are $x = 0$ and $x = 1$.

```
C     MAIN PROGRAM FOR ALL EQUILIBRIUM PROBLEMS AND FOR
C     THE FIRST MODE OF ALL EIGENVALUE PROBLEMS
C     M IS THE ORDER OF EQUATION AND N IS THE NUMBER
C     OF BOUNDARY CONDITIONS AT X=0
      DIMENSION R(100)
      COMMON M,N,IT,NC,NE,NP,MP1,NP1,EI
      COMMON B(8,8),D(8,16),C(100),E(10),P(100)
    1 FORMAT(4E16.8)
    2 FORMAT(5I5)
    3 FORMAT(1H1)
    4 FORMAT(/)
      READ(1,2) M,N,ITL
      MP1=M+1
      NP1=N+1
      CALL INITL
      WRITE(3,3)
      DO 40 IT=1,ITL
      CALL INTGRD
      CALL INTGRA(C,NC,R,NR,M)
      NC=NR
      DO 20 I=MP1,NC
   20 C(I)=R(I)
      CALL BOUND
      CALL MATMUL (D,M,C)
   40 WRITE(3,3)
      CALL RESULT
      STOP
      END
```

```
      SUBROUTINE BMTRX
C     THIS SUBROUTINE CONSTRUCTS D MATRIX FROM B MATRIX
C     FOR ALL EQUILIBRIUM PROBLEMS AND FOR THE
C     FIRST MODE OF ALL EIGENVALUE PROBLEMS
      DIMENSION F(10)
      COMMON M,N,IT,NC,AE,NP,MP1,NP1,EI
      COMMON B(8,8),D(8,16),C(100),E(10),P(100)
    1 FORMAT(4E16.8)
    2 FORMAT(14F5.1)
    3 FORMAT(//20X, 8HD MATRIX /)
      F(1)=1.
      DO 20 I=2,M
   20 F(I)=F(I-1)*FLOAT(I-1)
      READ(1,2)((B(I,J),J=1,MP1),I=1,M)
      DO 100 I=1,M
      IF(I.GT.N) GO TO 60
      DO 40 J=1,M
   40 D(I,J)=B(I,J)*F(J)
      GO TO 100
   60 D(I,1)=B(I,1)
      DO 80 J=2,M
      D(I,J)=B(I,1)
      DO 80 K=2,J
      JK=J-K
   80 D(I,J)=D(I,J)+F(J)*B(I,K)/F(JK+1)
  100 CONTINUE
      CALL INVRS(D,M)
      WRITE(3,3)
      WRITE(3,1) ((D(I,J),J=1,M),I=1,M)
      RETURN
      END
```

```
      SUBROUTINE INVRS(A,N)
C     THIS SUBROUTINE INVERTS MATRIX A OF ORDER N
C     BY GAUSS-SIEDEL REDUCTION
      DIMENSION A(8,16)
    2 FORMAT(2(//20X,10HNO INVERSE //))
C     AUGMENTING THE MATRIX A BY AN IDENTITY MATRIX
      NN=N+N
      DO 20 I=1,N
      IN=I+N
      DO 10 J=1,N
      JN=J+N
   10 A(I,JN)=0.
   20 A(I,IN)=1.
C     THE REDUCTION PROCESS STARTS HERE
      DO 100 M=1,N
   30 DIV=A(M,M)
      IF(DIV.EQ.0.0) GO TO 70
      DO 40 J=1,NN
   40 A(M,J)=A(M,J)/DIV
      DO 60 I=1,N
      IF(I.EQ.M) GO TO 60
      AIM=A(I,M)
      DO 50 J=1,NN
   50 A(I,J)=A(I,J)-AIM*A(M,J)
   60 CONTINUE
      GO TO 100
   70 DO 90 I=M,N
      IF(A(I,M).EQ.0.0) GO TO 90
C     DO LOOP 80 EFFECTS INTERCHANGE OF THE ITH AND MTH ROWS
      DO 80 J=1,NN
      DUMY=A(I,J)
      A(I,J)=A(M,J)
   80 A(M,J)=DUMY
      GO TO 30
   90 CONTINUE
      WRITE(3,2)
      GO TO 120
```

```
  100 CONTINUE
C     TRANSFERING INVERSE OF A IN THE PLACE OF A
      DO 110 I=1,N
      DO 110 J=1,N
      JN=J+N
  110 A(I,J)=A(I,JN)
  120 RETURN
      END
```

```
      SUBROUTINE BOUND
C     THE SUBROUTINE COMPUTES THE F VECTOR
C     FOR ALL EQUILIBRIUM PROBLEMS AND FOR THE
C     FIRST MODE OF ALL EIGENVALUE PROBLEMS
      DIMENSION Y(10)
      COMMON M,N,IT,NC,NE,NP,MP1,NP1,EI
      COMMON B(8,8),D(8,16),C(100),E(10),P(100)
    1 FORMAT(4E16.8)
    2 FORMAT(//20X,5HBOUND/)
      DO 10 I=1,M
   10 C(I)=B(I,NP1)
      CON=1.
      DO 30 I=1,M
      Y(I)=0.
      AI=I
      DO 20 JM=MP1,NC
      AJM=JM
      Y(I)=Y(I)+CON*C(JM)
   20 CON=CON*(AJM/(AJM+1.-AI))
      CON=1.
      DO 30 J=1,I
      AJM=M+1-J
   30 CON=CON*AJM
      DO 40 I=NP1,M
      DO 40 J=1,M
   40 C(I)=C(I)-B(I,J)*Y(J)
      WRITE(3,2)
      WRITE(3,1)(C(I),I=1,M)
      RETURN
      END
```

```
      SUBROUTINE MATMUL (A,N,B)
C     THIS SUBROUTINE MULTIPLIES A SQUARE MATRIX A
C     OF ORDER N BY A COLUMN VECTOR B. THE ANSWER
C     APPEARS IN THE COLUMN B.
      DIMENSION A(8,16),B(8),D(8)
    1 FORMAT(4E16.8)
    3 FORMAT(20X,6HMATMUL)
      WRITE(3,3)
      DO 100 I=1,N
      D(I)=0.
      DO 100 J=1,N
  100 D(I)=D(I)+A(I,J)*D(J)
      DO 200 I=1,N
  200 B(I)=D(I)
      WRITE(3,1)(B(I),I=1,N)
      RETURN
      END
```

```
      SUBROUTINE SEMUL (A,NA,B,NB,C,NC)
C     SERIES A * SERIES B = SERIES C
      DIMENSION A(100),B(100),C(100)
    1 FORMAT(4E16.8)
    3 FORMAT(/20X,21HSERIES MULTIPLICATION  /)
      WRITE(3,3)
      NC=NA+NB-1
      IF(NA.GT.NB) GO TO 101
      DO 100 I=1,NC
      MI=I
      C(I)=0.
      IF(I.GT.NA)  MI=NA
      DO 100 J=1,MI
      MIJ=I-J+1
  100 C(I)=C(I)+A(J)*B(MIJ)
      GO TO 201
  101 DO 200 I=1,NC
      MI=I
      C(I)=0.
      IF(I.GT.NB)  MI=NB
      DO 200 J=1,MI
      MIJ=I-J+1
  200 C(I)=C(I)+B(J)*A(MIJ)
  201 WRITE(3,1) (C(I),I=1,NC)
      RETURN
      END
```

```
      SUBROUTINE SEDIV (A,NA,B,NB,C,NC)
      DIMENSION A(100),B(100),C(100)
C     SUBROUTINE FOR DIVISION OF TWO SERIES
C     SERIES A/SERIES B=SERIES C
    1 FORMAT(4E16.8)
    3 FORMAT(//20X,6HSEDIV /)
      WRITE(3,3)
      I=1
      C(I)=A(I)/B(I)
   10 I=I+1
      MI=1
      IF(I.GT.NB)  MI=NB
      C(I)=A(I)
      DO 20 J=2,MI
      MIJ=I+1-J
   20 C(I)=C(I)-B(J)*C(MIJ)
      C(I)=C(I)/B(1)
      IF(I.LT.NC) GO TO 10
      WRITE(3,1) (C(I),I=1,NC)
      RETURN
      END
```

```
      SUBROUTINE DIFFER(C,NC,D,ND,N)
C     NTH DERIVATIVE OF SERIES C = SERIES D
      DIMENSION C(100),D(100)
    1 FORMAT(4E16.8)
    3 FORMAT (/20X, 10 HDERIVATIVE  /)
      WRITE(3,3)
      AF=1.
      DO 10 I=2,N
      AI=I
   10 AF=AF*AI
      IF(N.EQ.1)  AF=1.
      DO 20 I=1,NC
      IN=I+N
      AI=I
      AIN=IN
      D(I)=C(IN)*AF
   20 AF=AF*AIN/AI
      ND=NC-N
      WRITE(3,1) (D(I),I= 1,ND)
      RETURN
      END
```

```
      SUBROUTINE INTGRA (C,NC,D,ND,N)
C     NTH INTIGRAL OF SERIES C = SERIES D
      DIMENSION C(100),D(100)
    1 FORMAT(4E16.8)
    3 FORMAT (//20X, 11 HINTIGRATION  /)
      WRITE(3,3)
      N1=N+1
      AF=1.
      DO 10 I=2,N
      AI=I
   10 AF=AF*AI
      IF(N.EQ.1) AF=1.
      ND=NC+N
      DO 20 I=1,ND
      IN=I+N
      AI=I
      AIN=IN
      D(IN)=C(I)/AF
   20 AF=AF*AIN/AI
      WRITE(3,1) (D(I),I=N1,NC)
      RETURN
      END
```

```
      SUBROUTINE RESULT
C     THIS SUBROUTINE GIVES THE RESULTS OF AN
C     EQUILIBRIUM PROBLEM
      COMMON M,N,IT,NC,NE,NP,MP1,NP1,EI
      COMMON B(8,8),D(8,16),C(100),E(10),P(100)
    1 FORMAT(4E16.8)
      WRITE(3,1) (C(I),I=1,NC)
      RETURN
      END
```

```
      SUBROUTINE RESULT
C     THIS SUBROUTINE GIVES THE RESULTS OF AN
C     EIGENVALUE  PROBLEM
      COMMON M,N,IT,NC,NE,NP,MP1,NP1,EI
      COMMON B(8,8),D(8,16),C(100),F(10),P(100)
    1 FORMAT(4E16.8)
    2 FORMAT(/)
      WRITE(3,1) EI
      WRITE(2,1) EI
      WRITE(3,2)
      DO 20 I=1,NC
      IF(C(I).NE.0.0) GO TO 40
   20 CONTINUE
   40 EI=C(I)
      DO 60 J=1,NC
   60 C(J)=C(J)/EI
      WRITE(3,1) (C(I),I=1,NC)
      WRITE(2,1) (C(I),I=1,NC)
      RETURN
      END
```

APPENDIX C

Description of Final Problem

This is a steady state heat-transfer problem. At steady state there are no variations with time and the energy equation for a fluid in laminar motion in a cylincrecal tube can be written as

$$\frac{1}{a}\frac{\partial}{\partial a}\left(a\frac{\partial T}{\partial a}\right) = \frac{2U_m}{\alpha}\left[1 - \left(\frac{a}{\alpha}\right)^2\right]\frac{\partial T}{\partial z} \qquad (1)$$

where T = temperature;

a = radial co-ordinate measured from the tube axis; a_o, tube radius

U_m = mean flow velocity

α = thermal diffusivity,

z = anial co-ordinate measured from the entrance of the pipe.

The equation reflects the following postulates:

1) The fluid is incompressible;
2) No heat is transferred by conduction in the z direction.
3) The velocity distribution is parabolic at all cross-sections and is maintained independent of the temperature (i.e. viscosity and density are not the functions of temperature);
4) Thermal conductivity and the product γc_p are independent of temperature.

The boundary conditions specified were that

1) the fluid and wall were uniformly at temperature T_i for $z \leq 0$,

i.e. $T = T_i, \quad z \leq 0$

and 2) the wall temperature varies linearly for $z > 0$.

$$T = T_i\left(1 + \frac{B}{a_o} z\right) \qquad a = a_o$$

The following dimensionless variables were defined.

$$x = \frac{a}{a_o}$$

$$\omega = \frac{z}{a_o^2} \frac{\alpha}{2U_m}$$

$$t = \frac{T - T_i}{T_i}$$

In terms of these dimensionless variables the formulation (1) would be

$$\frac{1}{x} \frac{\partial}{\partial x} \left(x \frac{\partial t}{\partial x}\right) = (1 - x^2) \frac{\partial t}{\partial \omega}$$

A product solution of the form

$$t = 1 - \quad (x) \qquad (\omega)$$

was employed to separate the variables. It was found that

$$W = e^{-\beta^2 \omega}$$

where $-\beta^2$ was the constant of separation. The equation for the y was

$$\frac{d^2y}{dx^2} + \frac{1}{x}\frac{dy}{dx} + \beta^2(1-x^2)y = 0$$

with the boundary conditions

$$\frac{dy}{dx} = 0 \qquad \text{at } x = 0$$

$$y = 0 \qquad \text{at } x = 1.$$

The computer program in its entirety is given in the following pages.

```
C     MAIN PROGRAM FOR ALL EQUILIBRIUM PROBLEMS AND FOR
C     THE FIRST MODE OF ALL EIGENVALUE PROBLEMS
C     M IS THE ORDER OF EQUATION AND N IS THE NUMBER
C     OF BOUNDARY CONDITIONS AT X=0
      DIMENSION R(100)
      COMMON M,N,IT,NC,NE,NP,MP1,NP1,EI
      COMMON B(8,8),D(8,16),C(100),E(10),P(100)
    1 FORMAT(4E16.8)
    2 FORMAT(5I5)
    3 FORMAT(1H1)
    4 FORMAT(/)
      READ(1,2) M,N,ITL
      MP1=M+1
      NP1=N+1
      CALL INITL
      WRITE(3,3)
      DO 40 IT=1,ITL
      CALL INTGRO
      CALL INTGRA(C,NC,R,NR,M)
      NC=NR
      DO 20 I=MP1,NC
   20 C(I)=R(I)
      CALL BOUND
      CALL MATMUL (D,P,C)
   40 WRITE(3,3)
      CALL RESULT
      STOP
      END
```

```
      SUBROUTINE INITL
C     THIS SUBROUTINE INITIALIZES THE PROBLEM OF
C     HEAT TRANSFER TO AN INCOMPRESSIBLE FLUID
C     IN LAMINAR MOTION
      COMMON M,N,IT,NC,NE,NP,MP1,NP1,EI
      COMMON B(8,8),D(8,16),C(100),E(10),P(100)
      CALL DATRX
      E(1)=-1.
      E(2)=0.
      E(3)=1.
      NE=3
      C(1)=1.
      C(2)=0.
      C(3)=-1.
      NC=3
      RETURN
      END
```

```
      SUBROUTINE DMTRX
C     THIS SUBROUTINE CONSTRUCTS D MATRIX FROM B MATRIX
C     FOR ALL EQUILIBRIUM PROBLEMS AND FOR THE
C     FIRST MODE OF ALL EIGENVALUE PROBLEMS
      DIMENSION F(10)
      COMMON M,N,IT,NC,NC,NP,MP1,NP1,EI
      COMMON B(8,9),D(8,16),C(100),E(10),P(100)
    1 FORMAT(4E16.8)
    2 FORMAT(14F5.1)
    3 FORMAT(//20X, 8HD MATRIX /)
      F(1)=1.
      DO 20 I=2,M
   20 F(I)=F(I-1)*FLOAT(I-1)
      READ(1,2)((B(I,J),J=1,MP1),I=1,M)
      DO 100 I=1,M
      IF(I.GT.N) GO TO 60
      DO 40 J=1,M
   40 D(I,J)=B(I,J)*F(J)
      GO TO 100
   60 D(I,1)=B(I,1)
      DO 80 J=2,M
      D(I,J)=B(I,1)
      DO 80 K=2,J
      JK=J-K
   80 D(I,J)=D(I,J)+F(J)*B(I,K)/F(JK+1)
  100 CONTINUE
      CALL INVRS(D,M)
      WRITE(3,3)
      WRITE(3,1) ((D(I,J),J=1,M),I=1,M)
      RETURN
      END
```

```
      SUBROUTINE INVRS(A,N)
C     THIS SUBROUTINE INVERTS MATRIX A OF ORDER N
C     BY GAUSS-SIEDEL REDUCTION
      DIMENSION A(6,12)
    2 FORMAT(2(//20X,10HNO INVERSE //))
C     AUGMENTING THE MATRIX A BY AN IDENTITY MATRIX
      NN=N+N
      DO 20 I=1,N
      IN=I+N
      DO 10 J=1,N
      JN=J+N
   10 A(I,JN)=0.
   20 A(I,IN)=1.
C     THE REDUCTION PROCESS STARTS HERE
      DO 100 M=1,N
   30 DIV=A(M,M)
      IF(DIV.EQ.0.0) GO TO 70
      DO 40 J=1,NN
   40 A(M,J)=A(M,J)/DIV
      DO 60 I=1,N
      IF(I.EQ.M) GO TO 60
      AIM=A(I,M)
      DO 50 J=1,NN
   50 A(I,J)=A(I,J)-AIM*A(M,J)
   60 CONTINUE
      GO TO 100
   70 DO 90 I=M,N
      IF(A(I,M).EQ.0.0) GO TO 90
C     DO LOOP 80 EFFECTS INTERCHANGE OF THE ITH AND MTH ROWS
      DO 80 J=1,NN
      DUMY=A(I,J)
      A(I,J)=A(M,J)
   80 A(M,J)=DUMY
      GO TO 30
   90 CONTINUE
      WRITE(3,2)
      GO TO 120
```

```
  100 CONTINUE
C     TRANSFERING INVERSE OF A IN THE PLACE OF A
      DO 110 I=1,N
      DO 110 J=1,N
      JN=J+N
  110 A(I,J)=A(I,JN)
  120 RETURN
      END
```

```
      SUBROUTINE INTGRL
C     THIS SUBROUTINE NORMALIZES THE ITERATE,
C     FINDS ITS RALEIGHS QUOTIENT AND FORMS
C     THE INTIGRAND FOR THE PROBLEM OF
C     HEAT TRANSFER TO AN INCOMPRESSIBLE FLUID
C     IN LAMINAR MOTION
      DIMENSION F(100),G(100),H(100),R(100)
      COMMON M,N,IT,NC,NE,NF,MP1,NP1,EI
      COMMON R(8,8),D(8,16),C(100),F(10),P(100)
    1 FORMAT(4E16.8)
    2 FORMAT(/)
C     NORMALIZATION OF THE ITERATE
      DO 20 I=1,NC
      IF(C(I).NE.0.0) GO TO 30
   20 CONTINUE
   30 CIJ=C(I)
      DO 40 J=1,NC
   40 C(J)=C(J)/CIJ
      WRITE(3,1) (C(I),I=1,NC)
      WRITE(3,2)
C     CALCULATION OF RALEIGHS QUOTIENT
      CALL SEMUL(C,NC,E,NE,G,NG)
      WRITE(3,2)
      CALL SEMUL(C,NC,G,NG,H,NH)
      WRITE(3,2)
      F(1)=0.
      CALL INTGRA(H,NH,F,NF,1)
      WRITE(3,2)
      DN=0.
      DO 80 I=1,NF
   80 DN=DN+F(I)
      CALL DIFFER(C,NC,F,NF,1)
      WRITE(3,2)
      CALL DIFFER(F,NF,H,NH,1)
      WRITE(3,2)
      DO 60 I=1,NH
```

```
   60 H(I)=H(I)+F(I+1)
      WRITE(3,1) (H(I),I=1,NH)
      WRITE(3,2)
      CALL SEMUL (C,NC,H,NH,R,NR)
      WRITE(3,2)
      H(1)=0.
      CALL INTGRA (R,NR,H,NH,1)
      WRITE(3,2)
      DM=0.
      DO 70 I=1,NH
   70 DM=DM+H(I)
      H(1)=0.
      EI=DM/DN
      WRITE(3,1) EI,DM,DN
      WRITE(3,2)
C     CONSTRUCTION OF THE INTEGRAND
      NC=NG
      DO 90 I=1,NC
      C(I)=G(I)-F(I+1)/EI
      IF(I.GT.NF)  C(I)=G(I)
   90 CONTINUE
      R(1)=0.
      WRITE(3,1) (C(I),I=1,NC)
      WRITE(3,2)
      RETURN
```

```
      SUBROUTINE BOUND
C     THE SUBROUTINE COMPUTES THE F VECTOR
C     FOR ALL EQUILIBRIUM PROBLEMS AND FOR THE
C     FIRST MODE OF ALL EIGENVALUE PROBLEMS
      DIMENSION Y(10)
      COMMON M,N,IT,NC,NE,NP,MP1,NP1,EI
      COMMON B(8,8),C(8,16),C(100),E(10),P(100)
    1 FORMAT(4E16.8)
    2 FORMAT(//20X,5HBOUND/)
      DO 10 I=1,M
   10 C(I)=B(I,MP1)
      CON=1.
      DO 30 I=1,M
      Y(I)=0.
      AI=I
      DO 20 JM=MP1,NC
      AJM=JM
      Y(I)=Y(I)+CON*C(JM)
   20 CON=CON*(AJM/(AJM+1.-AI))
      CON=1.
      DO 30 J=1,I
      AJM=M+1-J
   30 CON=CON*AJM
      DO 40 I=NP1,M
      DO 40 J=1,M
   40 C(I)=C(I)-B(I,J)*Y(J)
      WRITE(3,2)
      WRITE(3,1)(C(I),I=1,M)
      RETURN
      END
```

```
      SUBROUTINE MATMUL (A,N,B)
C     THIS SUBROUTINE MULTIPLIES A SQUARE MATRIX A
C     OF ORDER N BY A COLUMN VECTOR B. THE ANSWER
C     APPEARS IN THE COLUMN B.
      DIMENSION A(8,16),B(8),D(8)
    1 FORMAT(4E16.8)
    3 FORMAT(20X,6HMATMUL)
      WRITE(3,3)
      DO 100 I=1,N
      D(I)=0.
      DO 100 J=1,N
  100 D(I)=D(I)+A(I,J)*B(J)
      DO 200 I=1,N
  200 B(I)=D(I)
      WRITE(3,1)(B(I),I=1,N)
      RETURN
      END
```

```
      SUBROUTINE SEMUL (A,NA,B,NB,C,NC)
C     SERIES A * SERIES B = SERIES C
      DIMENSION A(100),B(100),C(100)
    1 FORMAT(4E16.8)
    3 FORMAT(/20X,21HSERIES MULTIPLICATION  /)
      WRITE(3,3)
      NC=NA+NB-1
      IF(NA.GT.NB) GO TO 101
      DO 100 I=1,NC
      MI=I
      C(I)=0.
      IF(I.GT.NA)  MI=NA
      DO 100 J=1,MI
      MIJ=I-J+1
  100 C(I)=C(I)+A(J)*B(MIJ)
      GO TO 201
  101 DO 200 I=1,NC
      MI=I
      C(I)=0.
      IF(I.GT.NB)  MI=NB
      DO 200 J=1,MI
      MIJ=I-J+1
  200 C(I)=C(I)+B(J)*A(MIJ)
  201 WRITE(3,1) (C(I),I=1,NC)
      RETURN
      END
```

```
      SUBROUTINE DIFFER(C,NC,D,ND,N)
C     NTH DERIVATIVE OF SERIES C = SERIES D
      DIMENSION C(100),D(100)
    1 FORMAT(4E16.8)
    3 FORMAT (/20X, 10 HDERIVATIVE  /)
      WRITE(3,3)
      AF=1.
      DO 10 I=2,N
      AI=I
   10 AF=AF*AI
      IF(N.EQ.1)  AF=1.
      DO 20 I=1,NC
      IN=I+N
      AI=I
      AIN=IN
      D(I)=C(IN)*AF
   20 AF=AF*AIN/AI
      ND=NC-N
      WRITE(3,1) (D(I),I= 1,ND)
      RETURN
      END
```

```
      SUBROUTINE INTGRA (C,NC,D,ND,N)
C     NTH INTIGRAL OF SERIES C = SERIES D
      DIMENSION C(100),D(100)
    1 FORMAT(4E16.8)
    3 FORMAT (/20X, 11 HINTIGRATION  /)
      WRITE(3,3)
      N1=N+1
      AF=1.
      DO 10 I=2,N
      AI=I
   10 AF=AF*AI
      IF(N.EQ.1) AF=1.
      ND=NC+N
      DO 20 I=1,ND
      IN=I+N
      AI=I
      AIN=IN
      D(IN)=C(I)/AF
   20 AF=AF*AIN/AI
      WRITE(3,1) (D(I),I=N1,ND)
      RETURN
      END
```

```
      SUBROUTINE RESULT
C     THIS SUBROUTINE GIVES THE RESULTS OF AN
C     EIGENVALUE  PROBLEM
      COMMON M,N,IT,NC,NC,NP,MP1,NP1,EI
      COMMON B(8,8),D(8,16),C(100),E(10),P(100)
    1 FORMAT(4E16.8)
    2 FORMAT(/)
      WRITE(3,1) EI
      WRITE(2,1) EI
      WRITE(3,2)
      DO 20 I=1,NC
      IF(C(I).NE.0.0) GO TO 40
   20 CONTINUE
   40 EI=C(I)
      DO 60 J=I,NC
   60 C(J)=C(J)/EI
      WRITE(3,1) (C(I),I=1,NC)
      WRITE(2,1) (C(I),I=1,NC)
      RETURN
      END
```

ACKNOWLEDGEMENTS

I wish to express my thankfulness to Dr. Hugh S. Walker, for his guidance and counsel throughout the course of this work. In addition to being my Major Advisor, he has been my sourse of inspiration and guide for the computer work without which the present work would have been impossible.

I also wish to express my deep gratitude to Dr. Ralph G. Nevins, Head of the Department of Mechanical Engineering, for the financial assistance which aided in making the study possible, and for the careful review he has given to this paper.

VITA

Vasant M. Pandav

Candidate for the Degree of

Master of Science

Thesis: AN ITERATIVE TECHNIQUE FOR AN ANALYTICAL SOLUTION TO TWO-POINT BOUNDARY PROBLEMS.

Major Field: Mechanical Engineering

Biographical:

Personal Data: Born at Jepa-Pimpri, District Bhir, Maharashtra, India, July 10, 1943, the son of Madhavrao G. and Godawaribai M. Pandav.

Education: Attended Middle-Cum-High School, Patoda, Maharashtra; graduated from Y. N. Vidyalaya, Ambajogai, Maharashtra in March 1958; attended Govt. Arts & Sc. College, Aurangabad, and College of Engineering Aurangabad; received the Bachelor of Engineering (Mechanical) degree from the Marathwada University, Aurangabad D., India, in June 1964, completed requirements for the Master of Science degree at Kansas State University, in November, 1966.

Professional Experience: Worked in the capacity of an Assistant Lecturer in Mechanical Engineering, at College of Engineering, Aurangabad D., India, from June 1964 to August 1965.

AN ITERATIVE TECHNIQUE FOR AN ANALYTICAL SOLUTION TO TWO-POINT BOUNDARY PROBLEMS

by

VASANT MADHAVRAO PANDAV

B. E., Marathwada University, Aurangabad, India, 1964

AN ABSTRACT OF A MASTER'S THESIS

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Mechanical Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1967

An iteration technique for solving two-point boundary problems was studied. The technique yields an analytical solution in the form of a power series in the independent variable. This type of technique had been previously applied to the problem of buckling of columns by L. Vianello and to the problem of critical speeds of rotating shafts by A. Stodola. Here the method was applied to a larger class of problems.

A variety of problems of both equilibrium and eigen-value type were selected for the study. The problems varied in the order of complexity of the differential equations governing them. Some of them were simple enough to have solutions in closed form. These solutions were expanded in power series so as to make the comparison direct, and therefore, easy. Others did not have closed form solutions so that it was necessary to solve them by some other method. The solutions thus obtained were used to form a basis for comparison.

In the case of eigenvalue problems the method led to the mode corresponding to the lowest eigenvalue. The process was then modified to extract modes corresponding to higher eigenvalues. The orthogonality condition was used for this purpose.

Lastly, a problem of practical importance was selected. The problem was typical of its class. The governing equation had a lower order derivative term (independent of eigenvalue) present with the highest order derivative term. The terms had non-constant coefficients. To the author's knowledge, a closed form

solution had not been obtained for this problem at that time. However, a numerical solution with very good accuracy was available. This solution was used for comparison with that obtained by the present method.